



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Inteligencia de negocio y análisis predictivo aplicado a la educación

Autor/es

CRISTINA LAENCINA FERNÁNDEZ

Director/es

ARTURO JAIME ELIZONDO

Facultad

Escuela de Máster y Doctorado de la Universidad de La Rioja

Titulación

Máster Universitario en Tecnologías Informáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2018-19



Inteligencia de negocio y análisis predictivo aplicado a la educación, de
CRISTINA LAENCINA FERNÁNDEZ
(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative
Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.
Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los
titulares del copyright.

Trabajo de Fin de Máster

Inteligencia de negocio y análisis predictivo aplicado a la educación

Autora

Cristina Laencina Fernández

Tutor: Arturo Jaime Elizondo

MÁSTER:

Máster en Tecnologías Informáticas (853M)

Escuela de Máster y Doctorado



**UNIVERSIDAD
DE LA RIOJA**

AÑO ACADÉMICO: 2018/2019

Índice

1. RESUMEN EN CASTELLANO Y EN INGLÉS: SÍNTESIS DEL TFM	2
2. INTRODUCCIÓN	3
2.1. Justificación y antecedentes	3
2.2. Objetivos	4
2.3. Herramientas utilizadas	5
3. DESARROLLO	7
3.1. Inteligencia de negocio	7
3.1.1. Estructura de la base de datos de la aplicación online	7
3.1.2. Análisis de los gráficos	9
3.1.3. Creación del datawarehouse	10
3.1.4. Proceso ETL	11
3.1.5. Implementación de los cuadros de mando	16
3.1.6. Cuadros de mando en la aplicación	19
3.2. Análisis predictivo	28
3.2.1. Proceso ETL para la predicción del éxito o fracaso	28
3.2.2. Creación del dataset	29
3.2.3. Creación del modelo	31
3.2.4. Implementación del análisis predictivo	34
3.2.5. Análisis predictivo en la aplicación	36
4. CONCLUSIONES	38
5. BIBLIOGRAFÍA	39

1. RESUMEN EN CASTELLANO Y EN INGLÉS: SÍNTESIS DEL TFM

Resumen

En la actualidad, los profesores de los laboratorios de la asignatura de base de datos disponen de una aplicación llamada AplicacionBD, a través de la cual los alumnos realizan las prácticas de la asignatura y tienen la posibilidad de realizar ejercicios de repaso con el fin de promover el autoaprendizaje de los estudiantes e intentar mejorar su rendimiento en la asignatura.

A través de esta aplicación, los profesores disponen de los datos para poder analizar la evolución de cada uno de los estudiantes, pero no disponen del tiempo necesario para ello.

Por ello se ha llevado a cabo la realización de este proyecto con la creación de una aplicación web compuesta por diferentes cuadros de mando y el análisis predictivo de las posibles notas finales de los alumnos, con el fin de ayudar a los profesores en la parte práctica de la asignatura a poder ver de forma visual y actualizada el seguimiento de los alumnos y poder probar diferentes estrategias con la intención de mejorar los resultados de la asignatura.

Abstract

At present, the teachers of the laboratories of the database subject have an application called AplicacionBD, through which the students perform the practices of the subject and have the possibility of carrying out review exercises in order to promote the self-learning of the students and try to improve their performance in the subject.

Through this application, teachers have the data to analyze the evolution of each of the students, but do not have the necessary time to do so.

For this reason, the realization of this project has been carried out with the creation of a web application composed of different dashboard and the predictive analysis of the possible final grades of the students, in order to help teachers in the practical part of the subject to be able to see in a visual and updated way the follow-up of the students and be able to try different strategies with the intention of improving the results of the subject.

2. INTRODUCCIÓN

2.1. Justificación y antecedentes

Los profesores de la asignatura de base de datos, perteneciente al primer curso del Grado de Ingeniería Informática, disponen de una aplicación llamada AplicacionBD, la cual es usada en los laboratorios de dicha asignatura.

Al principio, esta aplicación consistía en una herramienta a través de la cual los alumnos tenían acceso a los enunciados de los ejercicios de prácticas de los laboratorios, los resultados esperados por los ejercicios y la facilidad de entrega de los mismos.

En ella se almacenaba toda la información necesaria para la realización de las prácticas y los datos de las entregas de los alumnos.

A través de los datos almacenados, los profesores pudieron observar el gran porcentaje de alumnos que copiaban los ejercicios de las prácticas.

Se incorporó a la herramienta un módulo que permitía a los alumnos utilizar la aplicación no solo de forma obligatoria en las prácticas, sino también de forma opcional para la realización de ejercicios de repaso. El objetivo de este nuevo módulo era promover el auto aprendizaje de los estudiantes, con el fin de mejorar el rendimiento. Como consecuencia de esta incorporación se pudo constatar que el número de copias había descendido de forma significativa.

Los profesores han realizado el estudio, que se presenta en el artículo “The Effects of Adding Non-Compulsory Exercises to an Online Learning Tool on Student Performance and Code Copying” [1], donde se analiza la influencia en el aprendizaje de este nuevo módulo y donde se revela la relación entre los ejercicios de repaso que realizan los alumnos con el descenso de las copias y el aumento en el rendimiento en los exámenes de evaluación continua.

A los profesores les gustaría poder consultar en cualquier momento del curso, la evolución de los alumnos en relación con los ejercicios de repaso y las copias que realizan y poderlos clasificar respecto a estas dos medidas y poder ensayar diferentes estrategias para aumentar su rendimiento.

2.2. Objetivos

El propósito del proyecto es cubrir las necesidades actuales de los profesores descritas anteriormente, para ello se va a realizar un proyecto de inteligencia de negocio y de análisis predictivo.

Los objetivos por cumplir para la parte del proyecto de inteligencia de negocio son, por un lado, la creación de una aplicación con diferentes cuadros de mando que muestren información útil a los profesores para poder tomar decisiones con ella.

Y, por otro lado, el objetivo de análisis predictivo consiste en añadir a la aplicación un apartado que sea capaz de predecir las notas finales que los alumnos van a obtener en la asignatura.

Tal y como se puede observar, el proyecto se puede diferenciar en dos grandes objetivos, pero a su vez estos se pueden dividir en pequeños objetivos:

- Realización del estudio de los datos de los que se dispone actualmente y de la información que puede servir de ayuda a los profesores.
- Estudio y diseño de los gráficos que se van a desarrollar.
- Creación del proceso ETL indispensable para la creación de los cuadros de mando.
- Desarrollo e implementación de los cuadros de mando en la aplicación.
- Creación del modelo necesario para el análisis predictivo.
- Desarrollo e implementación de la predicción de las notas finales de los alumnos.

2.3. Herramientas utilizadas

Durante la realización del proyecto se han utilizado las siguientes herramientas:

- **Pentaho data integración (Kettle)** [2]. Se trata de una herramienta Extract, Transform and Load (ETL) que sirve para realizar la extracción de los datos de diferentes fuentes, la transformación de los datos obtenidos y la carga de estos en otra fuente de datos. La ejecución de esta herramienta y el diseño de las transformaciones y trabajos se realizado a través de su interfaz gráfica Spoon.
- **Angular** [3]. Es un framework de desarrollo para java script, el cual nos facilita el desarrollo de aplicaciones web Single Page Application (SPA) y nos proporciona herramientas para trabajar con los elementos de una web de forma sencilla. Uno de los propósitos de angular es separar de forma completa entre el front-end y el back-end. Una aplicación SPA es una aplicación de una sola página en la que navegar entre secciones y páginas de la misma y la carga de los datos, la lleva a cabo de forma dinámica realizando asíncronamente llamadas al servidor sin tener que recargar la página en ningún momento. La versión utilizada en el proyecto ha sido angular 6.
- **Higchart** [4]. Se trata de una librería Javascript que sirve para la creación de gráficos. Esta librería permite de forma sencilla e interactiva la inserción de una gran variedad de gráficas en nuestra aplicación web.
- **MySQL**. Es un sistema de gestión de base de datos.
- **Spring**. Es un framework que se compone de herramientas y utilidades para la creación de aplicaciones web para la parte del back-end. Uno de los módulos que ofrece spring son los servicios Representational State Transfer (REST) con Spring MVC, el cual se ha utilizado para la comunicación entre el front-end y el back-end.

- **Maven.** Es una herramienta para la gestión de proyectos de software, siendo una de sus principales ventajas la gestión de las dependencias. Maven se basa en un fichero Project Object Model (POM) en XML, en el cual se especifica todo lo que el proyecto necesita. Aparte de generar las dependencias necesarias, Maven también se encarga de compilar, empaquetar y ejecutar los tests.
- **Weka.** Es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Contiene herramientas para la preparación de datos, clasificación, regresión, agrupación, extracción de reglas de asociación y visualización. En este proyecto se han utilizado las herramientas que proporciona para la clasificación de los datos.

3. DESARROLLO

En el presente proyecto se pueden diferenciar dos partes: la parte de inteligencia de negocio, que consta de la creación de los cuadros de mando y de todo el proceso necesario para ello; y la parte de análisis predictivo, en el que se va a estimar la nota final que van a obtener los alumnos en la asignatura.

A continuación, se explica el desarrollo de cada una de las partes mencionadas anteriormente.

3.1. Inteligencia de negocio

Esta parte del proyecto está compuesta por la creación de cuadros de mando y por el proceso previo a la realización de estos, que consta del diseño de las gráficas a realizar, la creación del almacén de datos o data warehouse y el proceso ETL.

3.1.1. Estructura de la base de datos de la aplicación online

Tal y como se ha explicado anteriormente, los profesores disponen de una aplicación para los laboratorios de la asignatura de base de datos, en la que se almacena la siguiente información:

- Datos acceso de los usuarios (profesores y alumnos).
- Información personal de los alumnos (nombre, apellidos, DNI, cuasi).
- Datos de los ejercicios realizados a lo largo del curso como, por ejemplo, el enunciado, la dificultad, la solución, etc.
- Los ejercicios, tanto de repaso como los realizados en las prácticas por cada uno de los alumnos, almacenando también la respuesta entregada, la fecha, el número de intentos, si el resultado es correcto, etc.

El sistema gestor de base de datos en el que se almacena esta base de datos es MySQL. En este proyecto se trabajará con varios estados de esta base de datos correspondientes al final de los cursos académicos. En estos estados que se han puesto a disposición del proyecto se hizo desaparecer cualquier referencia a las personas a las que correspondían los datos (haciendo uso del algoritmo MD5).

En la figura 1 se muestra el diagrama de la base de datos de la herramienta online para practicar con SQL denominada AplicacionBD.

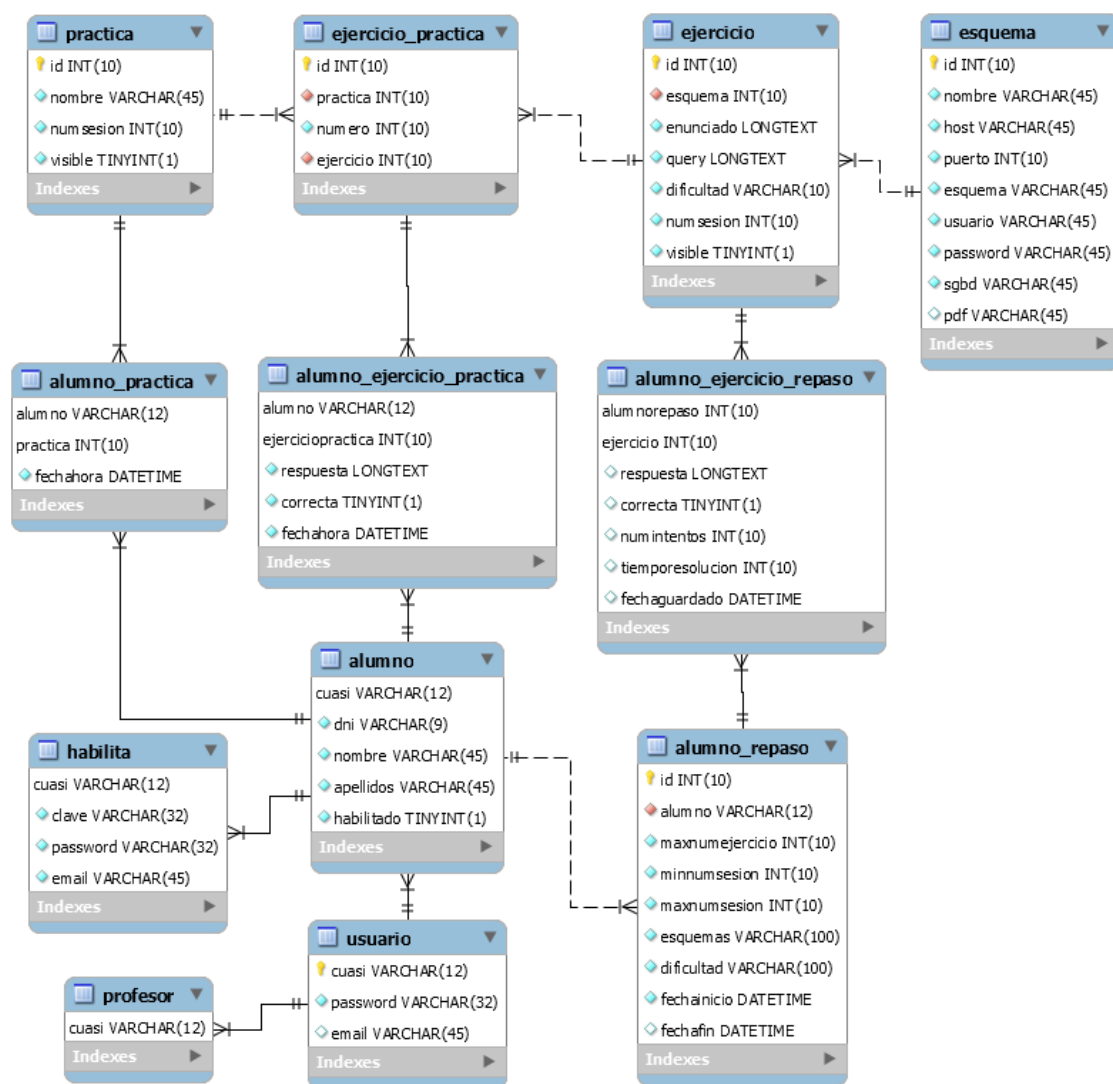


Figura 1. Diagrama base de datos aplicacionBD

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Prueb sesiones				TESTS			TOTAL	NO REC	EXAMEN	EXAMEN
2	Grp	Alum	1y2	3a6	7a9	"Entregas SQL"	T1	T2	T3	TESTS	TOTAL	MAYO	JULIO
3	INF-1	00a4	9,00	8,75	4,5	100,00	4,00	10,00	10,00	8,01	7,38		
4	INF-2	0121	5,00	4,75	10,00	67,5	6,00	2,00	6,68	4,89	7,01		
5	INF-3	0390	5,00	4,00	4,00	85,00	6,00	8,00	8,35	7,45	4,53		
6	INF-1	06dd	5,00	1,25	0,00	100,00	4,00	4,00	5,01	4,34	2,56		
7	INF-3	0949	5,00	3,5	7,00	80,00	8,00	6,00	1,67	5,22	5,13		
8	INF-1	0eaf	5,00	9,00	9,00	100,00	8,00	6,00	8,35	7,45	7,96		
9	INF-3	127e	10,00	9,5	8,00	100,00	6,00	6,00	6,68	6,23	8,37		
10	INF-3	1403	5,00	1,1	2,00	45,00	4,00			0	2,79		
11	INF-4	18a1	5,00	6,75	7,00	92,5	0,00	4,00	5,01	3	5,73		

Figura 2. Fichero Excel notas exámenes asignatura base de datos

Por otra parte, los profesores disponen de una hoja Excel, en la cual almacenan las notas de los alumnos en los distintos exámenes que se realizan

a lo largo de la asignatura. En las hojas que se facilitaron para la realización del proyecto también había desaparecido toda la información que pudiera identificar a los alumnos.

3.1.2. Análisis de los gráficos

El primer paso a realizar para la creación de los cuadros de mando es el estudio de los datos que serían interesantes para que los profesores puedan ver a través de las gráficas.

A la hora de realizar el estudio, hay que tener en cuenta que los datos que se obtengan de las gráficas tienen que servir a los profesores para poder identificar distintos grupos de alumnos, con el fin de planificar estrategias específicas para cada uno de ellos, al objeto de obtener mejores resultados.

Después de ver los datos de los que disponen los profesores y los posibles datos que se pueden calcular a partir de ellos, se ha llegado a la conclusión de que resultaría útil poder ver de forma rápida y sencilla la siguiente información.

En primer lugar, el porcentaje de los alumnos que copian más y de los que copian menos, al igual que poder consultar de forma más detallada información de los alumnos que se encuentran en cada rango.

También es importante para los profesores tener la posibilidad de ver el porcentaje de alumnos que realizan más repasos y los que realizan menos, e información detallada de los mismos.

Además, se plantea la opción de ver una relación entre las copias y los repasos y entre los ejercicios originales y los repasos.

Por último, es deseable poder observar a través de una gráfica quién copia a cada alumno.

Una vez realizado este análisis de lo que se pretende ver representado a través de las gráficas, se realiza otro análisis para determinar qué gráficas representan mejor cada uno de los datos a mostrar.

En conclusión, las gráficas vienen representadas mediante gráficas de barras, circulares, de burbujas y de líneas.

3.1.3. Creación del datawarehouse

Una vez realizado el estudio de las gráficas que se van a crear y los datos que son necesarios para ello, el siguiente paso a realizar es el diseño del datawarehouse.

Un datawarehouse es una base de datos que se caracteriza por integrar y depurar información de una o más fuentes de datos distintas. La información que se almacena en estas bases de datos debe ser homogénea y fiable.

El datawarehouse puede tener distintas estructuras, siendo las más utilizadas la estructura en estrella y la estructura en copo de nieve.

El modelo en estrella consta de una tabla de hechos y varias tablas de dimensiones, mientras que el modelo de copo de nieve es una variación del anterior, ya que las tablas de dimensiones se relacionan con otras tablas además de la tabla de hechos.

El modelo elegido en el proyecto es el modelo en estrella, pero en este caso, no solo disponemos de una tabla de hechos, sino que tenemos dos tablas de hechos y las tablas de dimensiones que se relacionan con estas.

La figura 3 contiene el diseño del datawarehouse de la aplicación.

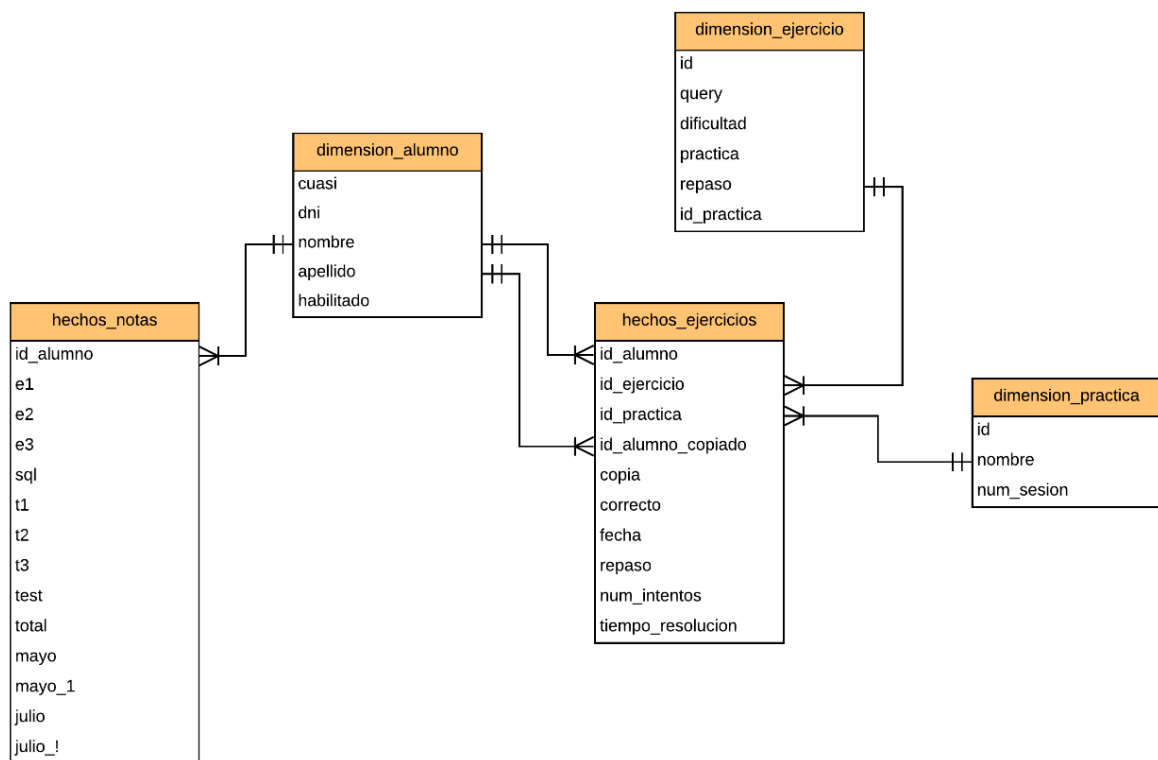


Figura 3. Diagrama de base de datos del datawarehouse

En el anterior diagrama puede verse que las tablas de hechos son la tabla hechos_notas y la tabla hechos_ejercicios que incluye tanto los ejercicios de repaso como de las prácticas.

En la tabla hechos_notas se almacenan las notas de los exámenes de evaluación continua (e1, e2 y e3), la nota final de las entregas de las prácticas (sql), los tests (t1, t2, t3), el total de los tests (test), la parte de la nota que no es recuperable de la evaluación (total, media ponderada de sql, e1, e2, e3 y test), el examen de mayo y acta de mayo (mayo, mayo_1), el examen de julio y acta de julio (julio, julio_1) y el id del alumno (denominado CUASI en la universidad).

En la tabla hechos_ejercicios se almacena, en el caso de tratarse de un ejercicio de repaso el id_alumno, id_ejercicio, si el ejercicio es correcto o no, la fecha de entrega, si se trata de un ejercicio de repaso (repaso), el número de intentos de realización del ejercicio y el tiempo de resolución, el resto de los campos se almacenan a nulo.

En cambio, en el caso de tratarse de un ejercicio de prácticas, se almacena lo mencionado anteriormente, indicando que no es un ejercicio de repaso (repaso) y si el ejercicio es copiado o no (copia) y en caso afirmativo, se almacena el id del alumno al que ha copiado.

En las tablas de dimensiones se almacenan los datos correspondientes al alumno, al ejercicio y a la práctica, tal y como se puede ver en el diagrama.

3.1.4. Proceso ETL

Una vez terminado el diseño y creado el datawarehouse, se comenzó el proceso ETL para cargarle los datos desde la base de datos de la herramienta Aplicación BD y desde la hoja de cálculo.

Para la realización de la ETL se utiliza la herramienta Kettle, mencionada en el apartado herramientas utilizadas.

En este proceso se ha tenido que llevar a cabo la extracción de los datos de dos fuentes distintas, la base de datos descrita anteriormente (ver Figura 1), que contiene los datos de los ejercicios de repaso y de las prácticas realizadas por los alumnos; la segunda fuente de datos es un fichero Excel, que contiene las notas de las diferentes pruebas realizadas por los alumnos.

Una vez hecha la extracción, se crearon distintas transformaciones de los datos antes de almacenarlos en el datawarehouse.

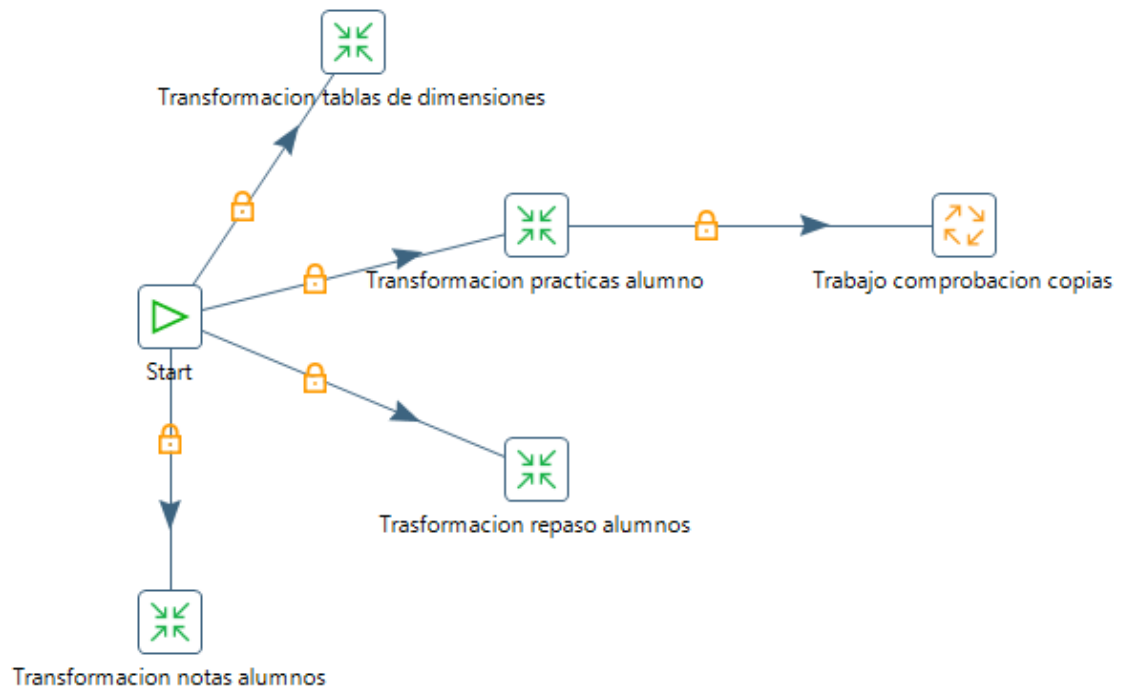


Figura 4. Trabajo principal ETL

Tal y como podemos ver en la figura 4, el trabajo principal consta de cuatro transformaciones distintas.

La “Transformación tabla de dimensiones” de la figura 5, hace referencia a la carga de los datos de las tablas de dimensiones de prácticas, ejercicios y alumnos.

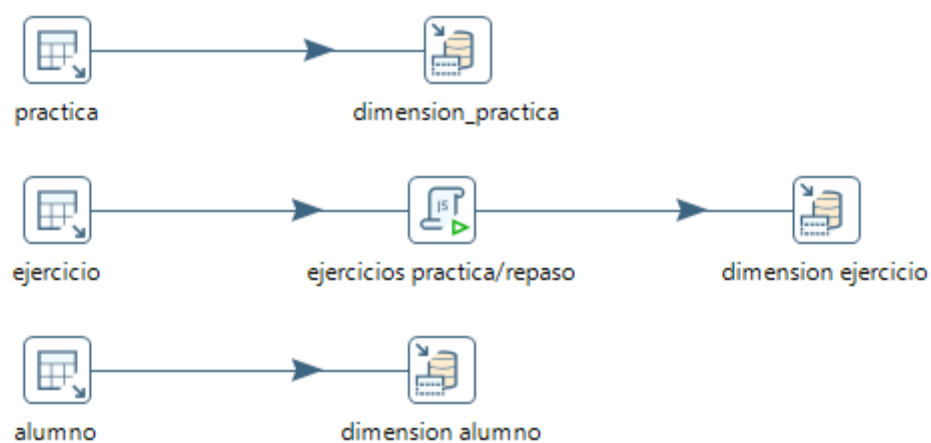


Figura 5. Transformación datos tablas dimensiones

Esta transformación es sencilla, ya que apenas precisa de modificaciones en los datos. Solo ha sido necesario diferenciar qué ejercicios corresponden a las prácticas y cuáles a ejercicios de repaso. Algún ejercicio podría pertenecer a ambos grupos.

La siguiente transformación, mostrada en la figura 6, corresponde con la carga de datos de los ejercicios que realizan los alumnos en las prácticas. Esta parte es la más compleja del proceso debido a las comprobaciones y transformaciones que se tienen que desarrollar.

En la BD de AplicacionBD no se almacena la información de si un ejercicio hecho en una práctica es copiado o no, por lo que los profesores, para comprobar si el ejercicio que un alumno ha entregado es copiado, lo comprueban en el momento de la consulta sin guardar el resultado.

Al necesitar la información sobre si los ejercicios son copiados y qué alumno ha desarrollado el ejercicio original, esta comprobación hay que efectuarla en el proceso ETL, ya que, si se realiza a la hora de obtener los datos para mostrarlos en las diferentes gráficas, este proceso ralentizaría bastantes las consultas.

Por esa razón, tal y como se refleja en la figura 4, esta parte está compuesta por una transformación y un trabajo secundario.

La “Transformación practica alumno” de la figura 6 corresponde con la obtención de los datos de los ejercicios de prácticas.

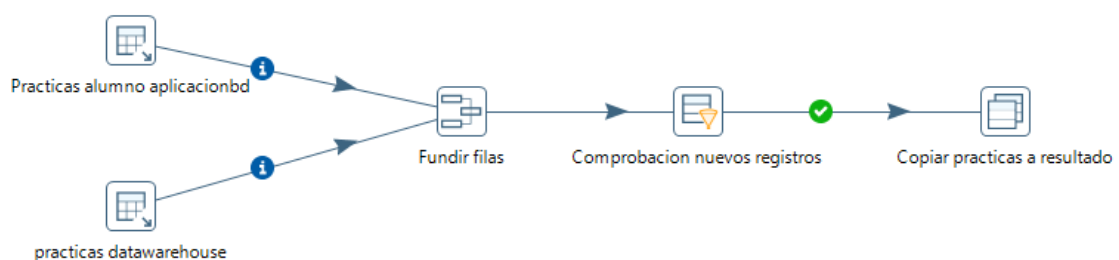


Figura 6. Transformación obtención datos ejercicios prácticas

Los datos obtenidos son “copiados” al trabajo secundario, en el cual se realiza la comprobación de si un ejercicio es copiado o no y, en caso afirmativo, se almacena el alumno del cual ha copiado. Esto se muestra en la figura 7.



Figura 7. Transformación comprobación copias ejercicios prácticas

Para que un ejercicio sea considerado copia, el resultado tiene que ser exactamente igual a otro, sin tener en cuenta los espacios o los signos de puntuación utilizados en la respuesta.

Si existen varios alumnos con el mismo resultado, se considera que la respuesta del alumno que haya entregado en primer lugar el ejercicio es la original y el resto son copias de este.

La siguiente transformación, representada en la figura 8, corresponde con los ejercicios de repaso que realizan los alumnos a lo largo del curso.

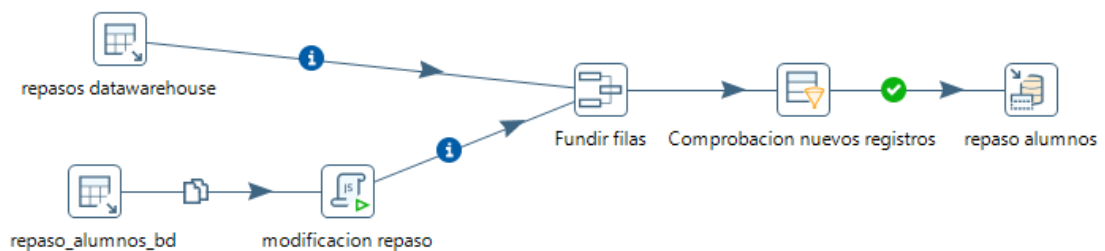


Figura 8. Transformación ejercicios de repaso

En este caso no es necesario efectuar ninguna modificación de los datos obtenidos.

Por último, se ejecuta la transformación de la figura 9 correspondiente a las notas de los alumnos. Los datos son obtenidos de un fichero Excel y almacenados en la tabla de hechos_notas, sin necesidad de realizar ninguna modificación intermedia.



Figura 9. Transformación notas alumnos

Como las prácticas se llevan a cabo cada semana, este proceso ETL tiene que ejecutarse después de terminar la última sesión de cada práctica para que la base de datos esté actualizada.

Tal y como se ha mencionado anteriormente, la información que contiene el datawarehouse debe ser fiable, por lo que hay que almacenar en el datawarehouse únicamente los datos que son nuevos.

Para ello, en todos los procesos explicados anteriormente se ha incluido una parte que se encarga de obtener los datos de la fuente original y del datawarehouse, y se unen esos dos resultados en el siguiente paso, que se denomina fundir filas (ver figura 8). Si la fila no se encuentra almacenada en el datawarehouse, marca un flag que indica que se trata de un nuevo registro, en cuyo caso se almacena la información.

En el único proceso en el que no se realiza esto es en las notas de los alumnos, ya que estas pueden ir cambiando.

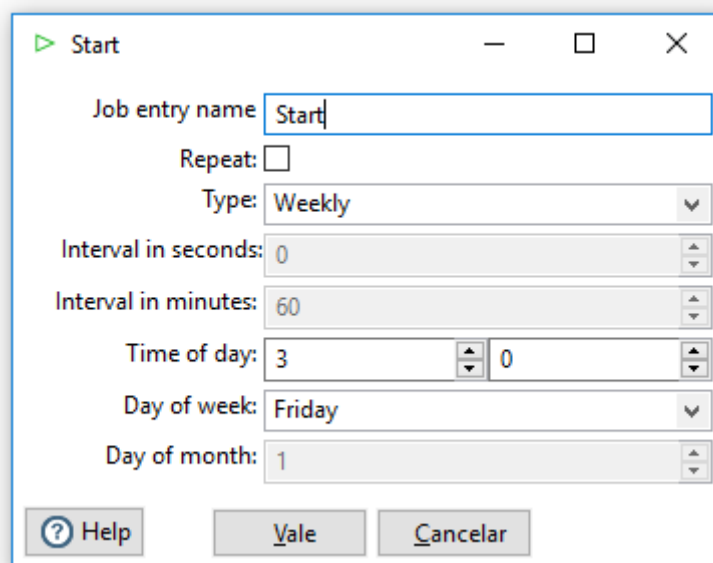


Figura 10. Programación ejecución proceso ETL

El proceso ETL (ver figura 10) debe ejecutarse una vez a la semana, porque en cada sesión de laboratorio se almacenan nuevos datos de los ejercicios de prácticas.

Para poder ejecutarlo cada semana, hay que programarlo para que se lance de forma automática.

La herramienta Kettle/Spoon permite realizar distintas programaciones:

- Por intervalos de minutos y segundos.

- Diariamente, en el que es posible indicar la hora exacta a la que se desee que el proceso se ejecute.
- Semanalmente, al igual que en el caso anterior, se puede especificar el día de la semana y la hora.
- Y, por último, mensualmente, indicando el día del mes y la hora.

En este caso, tal y como se ha mencionado anteriormente, la programación es semanal y coincidiendo con una hora y día en el que se prevea que la base de datos no será utilizada.

3.1.5. Implementación de los cuadros de mando

Después de finalizar el proceso ETL y tener todos los datos necesarios en el datawarehouse, se comenzó la implementación de las gráficas.

La aplicación está desarrollada mediante el patrón de modelo vista controlador. La capa de la vista, o front-end, se ha programado mediante el framework angular.

El front-end está compuesto especialmente por los gráficos programados con Higcharts, que precisan la instalación del paquete a través del sistema de gestión de paquetes NPM [5].

```
>npm install highcharts-angular --save
```

También hay que asegurarse de que se ha añadido correctamente el módulo en app.module.ts.

```
import { HighchartsChartModule } from 'highcharts-angular';

imports: [
  BrowserModule,
  AppRoutingModule,
  HighchartsChartModule,
```

A continuación, se instala el componente de Higcharts para después poder importarlo en nuestra aplicación.

```
>npm install highcharts --save
```

Una vez instalado todo lo necesario para poder utilizar Higcharts, se agrega la plantilla de Higchart en el html.

```
<highcharts-chart
[Highcharts]="HighchartsGraficoBarras"
[constructorType]="chartConstructor"
```

```
[options]="chartOptionsGraficoBarras"
[callbackFunction]="chartCallback"
[update]="updateFromInput2"
[oneToOne]="true"
```

>

Hay que considerar las siguientes variables

- **[Highcharts]** → esta variable es obligatoria, ya que es la que crea la instancia de Highcharts.
- **[constructorType]** → esta variable es opcional y el valor que se ha utilizado es 'chart', que es el constructor estándar de Highcharts.
- **[options]** → esta variable es obligatoria y es donde se configura el gráfico que se quiere representar con las opciones que se deseen.
- **[callbackFunction]** → esta variable es opcional, se trata de una función de devolución del gráfico creado, en este caso es this, que apunta al gráfico creado.
- **[update]** → esta variable es opcional y se trata de un booleano que sirve para poder actualizar el gráfico.
- **[oneToOne]** → esta variable es opcional, estableciendo el valor a true los datos de las series se actualizarán uno a uno.

El valor de las opciones explicadas anteriormente se configura en el componente que corresponda. A continuación, se muestra un ejemplo de ello [6].

```
HighchartsGraficoBarras = Highcharts;
chartConstructor = "chart";
chartCallback;
updateFromInput2 = false;

chartOptionsGraficoBarras = {
  chart: {
    type: "column"
  },
  title: {
    text: 'Alumnos que ' + this.rangoRepasos.toLowerCase()
  },
  xAxis: {
    categories: []
  },
  yAxis: {
    min: 0,
    title: {
      text: ""
    }
  },
  tooltip: {
    pointFormat:
      '<span style="color:{series.color}">{series.name}</span>:
      <b>{point.y}</b> ({point.percentage:.0f}%)<br/>',
    shared: true
  },
  plotOptions: {
    column: {
      stacking: "percent"
    }
  },
  series: []
};
```

```

constructor(private router: Router, private copiasService:
CopiasService) {
  const self = this;

  this.chartCallback = chart => {
    self.chart = chart;
  };
}

```

Para poder actualizar los datos, como se ha explicado anteriormente, el flag de la actualización se debe poner a true. Una vez actualizado, automáticamente el flag vuelve a tomar el valor false.

```

public actualizarDatosGrafBarras(data) {
  this.chartOptionsGraficoBarras.series = data.data;
  this.chartOptionsGraficoBarras.xAxis = {
    categories: data.alumnos
  };
  this.updateFromInput = true;
  this.chart.hideLoading();
}

```

Hay algunos gráficos que constan de partes más avanzadas y que precisan de la importación de otros componentes de angular.

```

addMore(Highcharts);
Drilldown(Highcharts);

```

Para obtener los datos de las gráficas y del resto de la aplicación, las peticiones se realizan a través de los servicios de angular a los controladores de la parte del back-end. Los datos que se envían son en formato JSON.

En los servicios de angular se mandan los parámetros necesarios al controlador, en este caso el filtro, mediante un método get.

```

public obtenerGraficoBarras(filtro) {
  return this.http.get<GrafBarras>({ url: this.userUrl + "repasoGrafBarras", options: {
    params: {
      filtro: JSON.stringify(filtro)
    },
    observe: 'response'
  }});
}

```

En el controlador se recogen los datos mandados desde el front-end, se convierten al data transfer object (DTO) correspondiente y se realizan los métodos necesarios para obtener el resultado deseado.

```

@RequestMapping(value= "/repasoGrafBarras", method= RequestMethod.GET)
public ResponseEntity<GraficaBarrasDto> graficaBarrasRepaso(@RequestParam(name = "filtro", required = false)
String filtroString){
    try {
        Filtro filtro = new ObjectMapper().readValue(filtroString, Filtro.class);
        GraficaBarrasDto graficoBarra = repasoService.obtenerGraficoBarrasRepasoAlumnoPorTramos(filtro);
        return new ResponseEntity<GraficaBarrasDto>(graficoBarra, HttpStatus.OK);
    } catch (IOException e) {
        return new ResponseEntity<GraficaBarrasDto>((GraficaBarrasDto) null, HttpStatus.EXPECTATION_FAILED);
    }
}

```

3.1.6. Cuadros de mando en la aplicación

En este apartado se expone detalladamente cada uno de los menús y gráficas de la aplicación. El menú principal se muestra en la figura 11.

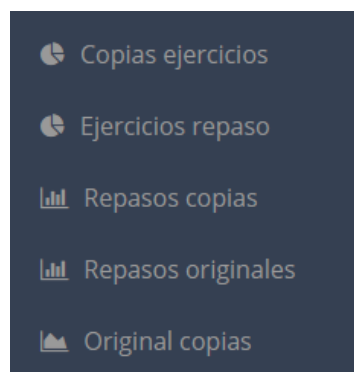


Figura 11. Menú aplicación

La primera opción del menú es “Copias ejercicios”, en la cual puede verse el porcentaje de alumnos que copian por tramos e información más detallada de estos alumnos en referencia a los ejercicios realizados en las prácticas. Tal y como podemos observar en la figura 12, esta pantalla está compuesta por un filtro que se encuentra en la zona superior y dos gráficos en la parte central.

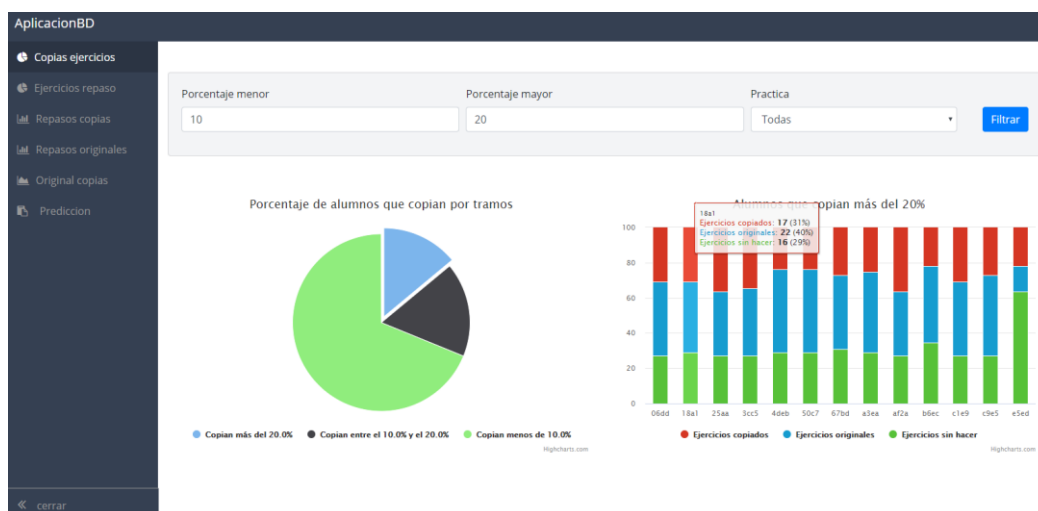


Figura 12. Diseño menú copias ejercicios

En la parte de la izquierda se encuentra un gráfico circular que representa el porcentaje de alumnos que copian por tramos.

Por defecto los tres grupos de la gráfica circular de la figura 13 hacen referencia a los alumnos que copian menos de un 5% de los ejercicios totales de las prácticas, los que copian más de un 15% y los que copian entre un 5% y un 15%.

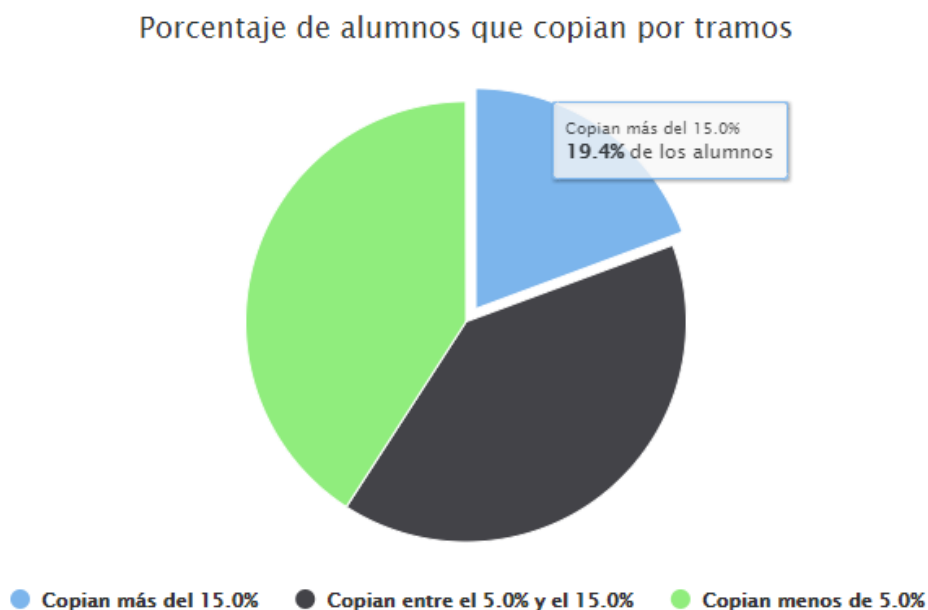


Figura 13. Gráfico circular del porcentaje de los alumnos que copian por rangos

Estos porcentajes se pueden cambiar mediante el filtro superior, indicando el porcentaje menor y el porcentaje mayor.

En los filtros, además de los porcentajes explicados anteriormente, se puede indicar una práctica, para poder mostrar solo los porcentajes de los alumnos que copian en una determinada práctica.

El gráfico de barras que se encuentra en la parte de la derecha (ver figura 12 y 14) está relacionado con el gráfico circular. En este gráfico de barras, puede apreciarse cuántos ejercicios originales y copiados han entregado los alumnos y cuántos no hay entregado.

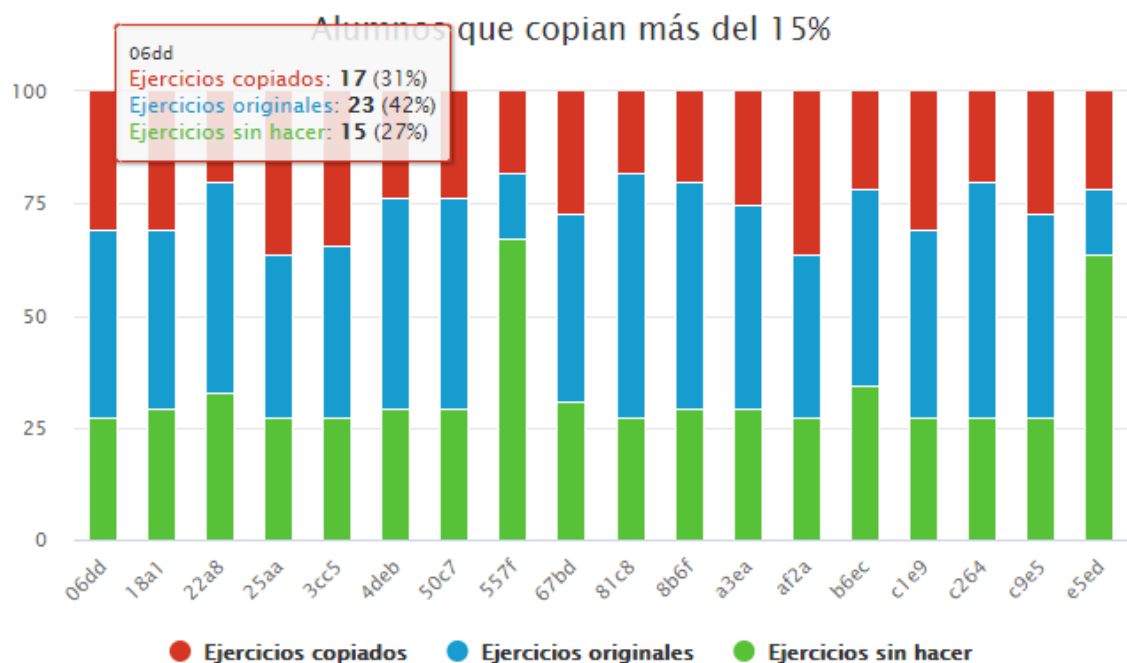


Figura 14. Gráfico barras de alumnos pertenecientes a un rango determinado de copias

La siguiente opción del menú principal es “Ejercicios repaso”. En esta pantalla, se presentan los ejercicios de repaso realizados hasta el momento por los estudiantes (ver figura 15).

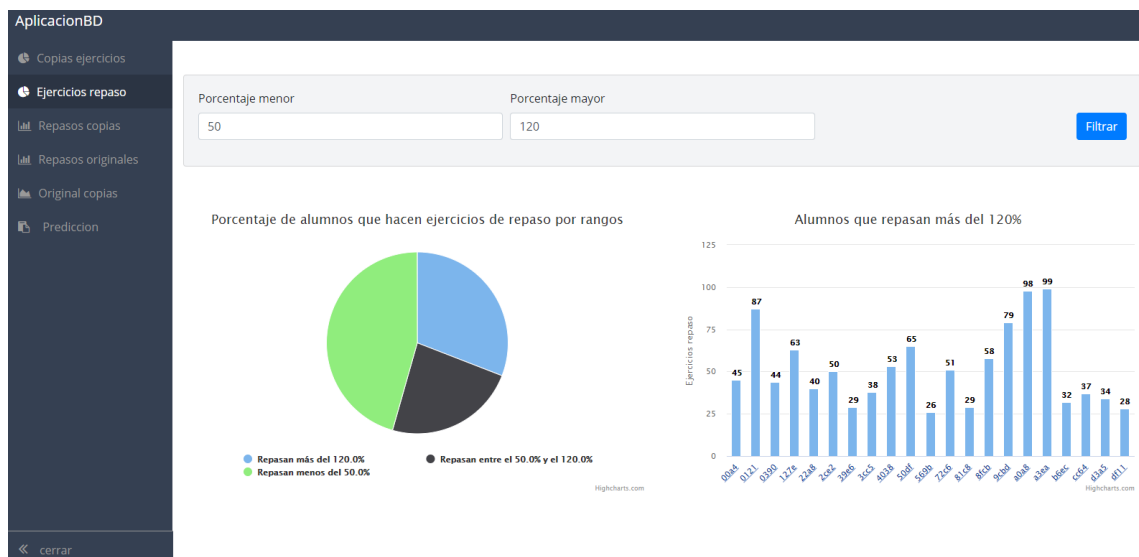


Figura 15. Diseño menú ejercicios repaso

En esta ocasión esta opción de menú está compuesta por los filtros en la parte superior y las dos gráficas en la parte central siempre visibles y una tercera gráfica que se ve en función de la interacción con el gráfico de barras.

Por uniformidad con la gráfica anterior, se ha incluido un filtro con dos porcentajes. En este caso, el porcentaje no se refiere al número total de ejercicios de repaso, sino al número medio de ejercicios de repaso correctos entregados por los alumnos, por lo que, a diferencia de la anterior gráfica, en este caso es posible superar el 100% en el filtro ya que hay alumnos que realizan más ejercicios que la media.

Inicialmente los rangos serán los alumnos que repasan menos de un 50%, los que repasan más del 120% y los alumnos que se encuentran entre el 50% y el 120%, pero, al igual que en la anterior gráfica, estos porcentajes se pueden modificar a través de los filtros de porcentaje menor y porcentaje mayor.

Lo explicado anteriormente se representa mediante el gráfico circular situado en la parte izquierda de la pantalla (ver figura 16).

Porcentaje de alumnos que hacen ejercicios de repaso por rangos

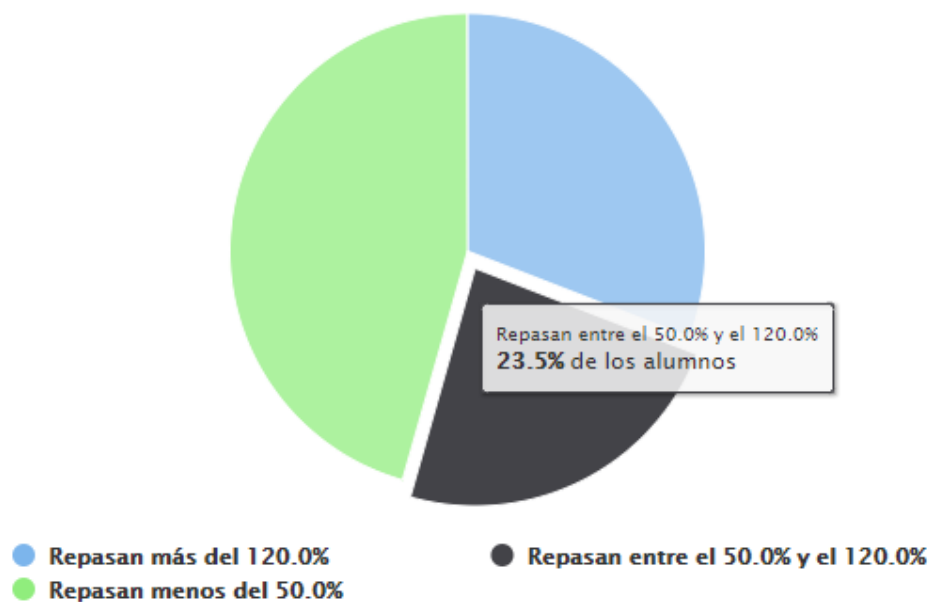


Figura 16. Gráfico circular del porcentaje de alumnos que hacen ejercicios de repaso por rangos

El gráfico de barras situado a la derecha en la figura 15 está relacionado con el gráfico circular, ya que muestra el número de ejercicios que ha realizado cada uno de los alumnos que se encuentran en un determinado rango.

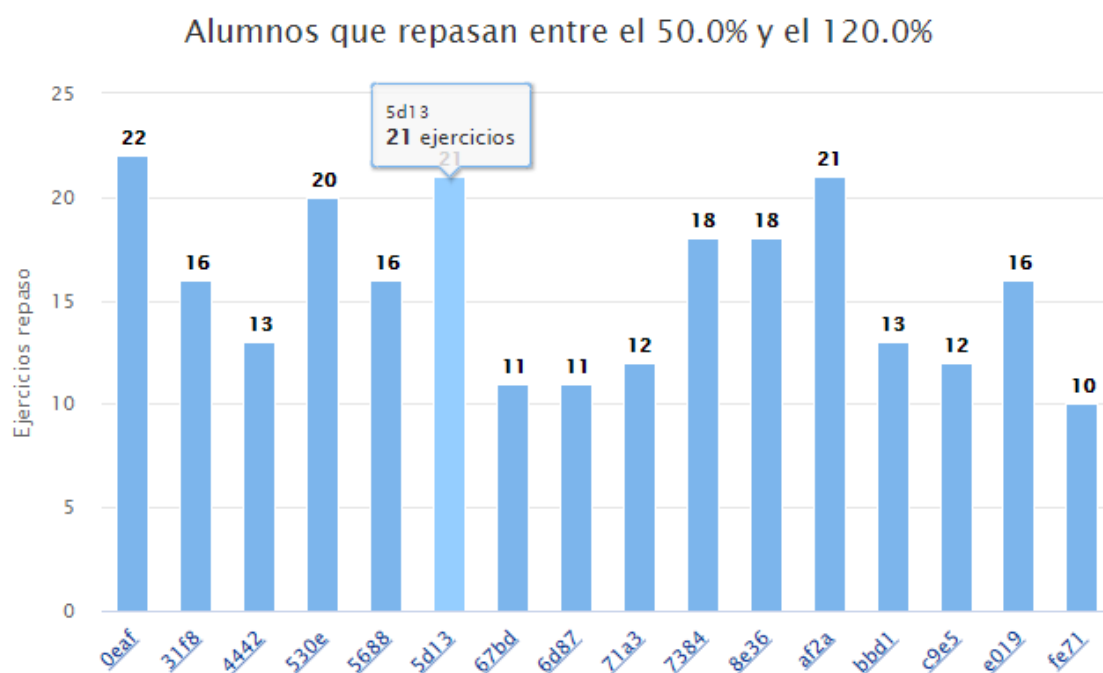


Figura 17. Gráfico barras de los alumnos pertenecientes a un rango determinado de ejercicios realizados de repaso

Seleccionando cualquier barra de la figura 17 se muestra cuántos ejercicios ha realizado el alumno de cada nivel de dificultad (difícil, fácil y medio) como se muestra en la figura 18.

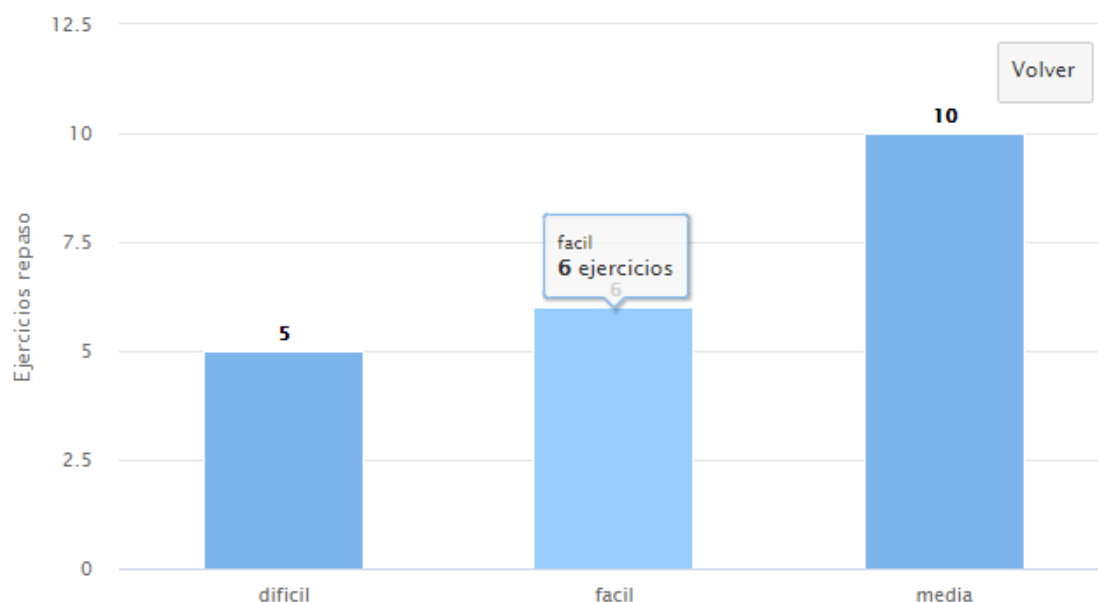


Figura 18. Gráfico barras de ejercicios realizados por un alumno

Además de lo anterior, se muestra en qué momento del curso ha realizado ejercicios el alumno seleccionado.

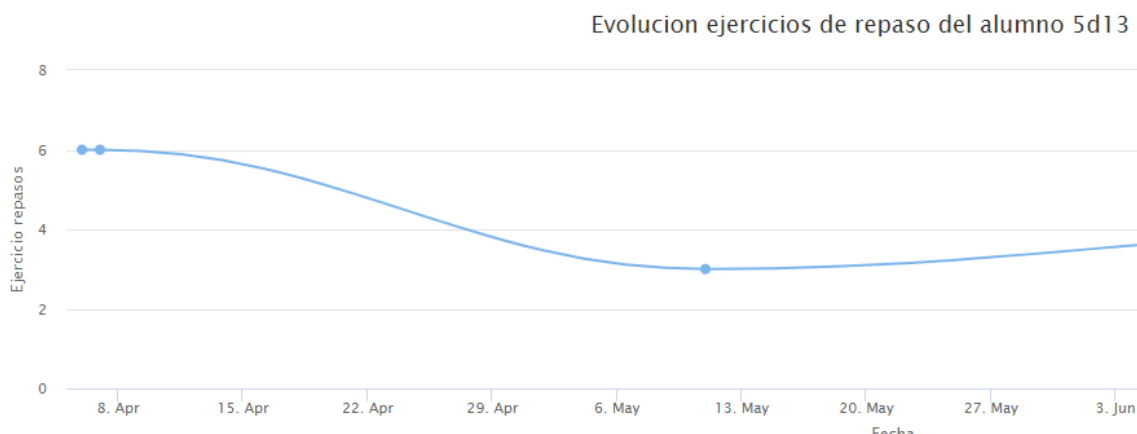


Figura 19. Evolución de los ejercicios de repaso realizados por un alumno

Esta última grafica puede resultar de mucha utilidad para los profesores ya que, como se explica en el artículo “The Effects of Adding Non-Compulsory Exercises to an Online Learning Tool on Student Performance and Code Copying” [1], existe una relación entre el momento del curso en el que realizan los repasos y las notas que van obteniendo en las pruebas.

La tercera opción del menú principal es “Repasos copias”, en el que se ve reflejada la relación entre los repasos y las copias de los ejercicios de prácticas.

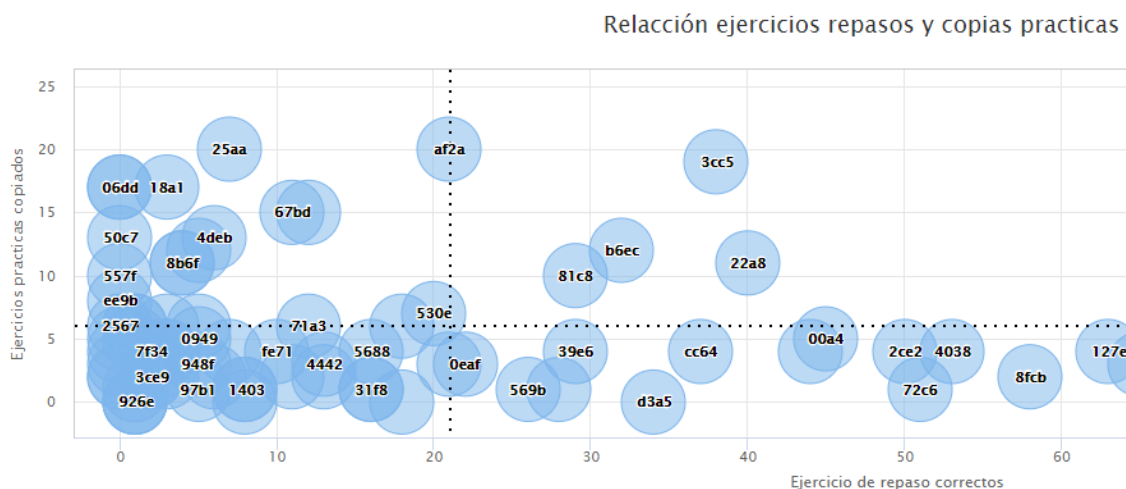


Figura 20. Gráfico de burbujas de la relación entre copias y repasos

A través de estas gráficas (ver figura 20), los profesores pueden identificar qué alumnos repasan y copian, y les puede permitir diseñar estrategias que mejoren el rendimiento de estos alumnos. Además de poder observar qué

alumnos repasan y copian se puede ver quienes progresan mejor en la asignatura.

La gráfica de la figura 20 incluye dos líneas que marcan la media de los ejercicios de repaso y la media de las prácticas copiadas y, de esta forma, los estudiantes quedan clasificados según la posición que ocupan en la gráfica.

La siguiente opción del menú principal es “Repasos originales”, y la gráfica que ofrece (ver figura 21) es muy parecida a la anterior, pero en este caso muestra la relación entre ejercicios de repaso realizados y ejercicios originales (sin copiar) entregados en las prácticas. La gráfica también contiene dos líneas que marcan la media de ejercicios de repaso realizados y la media de ejercicios originales.

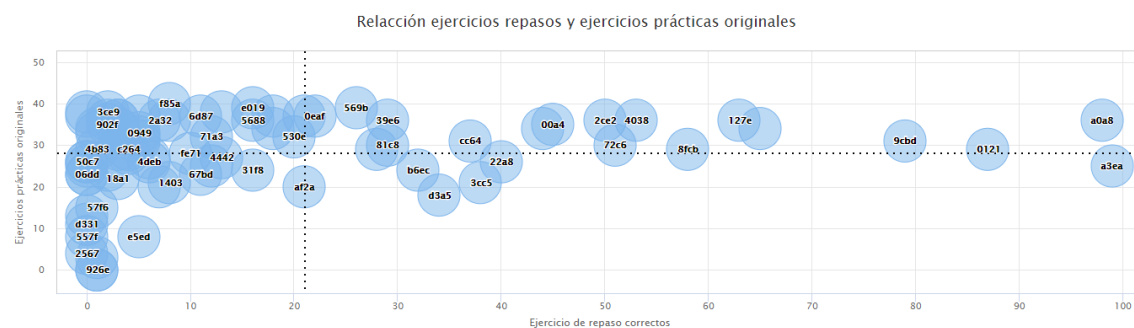


Figura 21. Gráfico de burbujas que relaciona ejercicios originales y repasos

Las dos gráficas anteriores permiten hacer un zoom sobre una zona específica para verla mejor. Esto es especialmente interesante cuando se concentran juntos bastantes alumnos (ver figura 22).

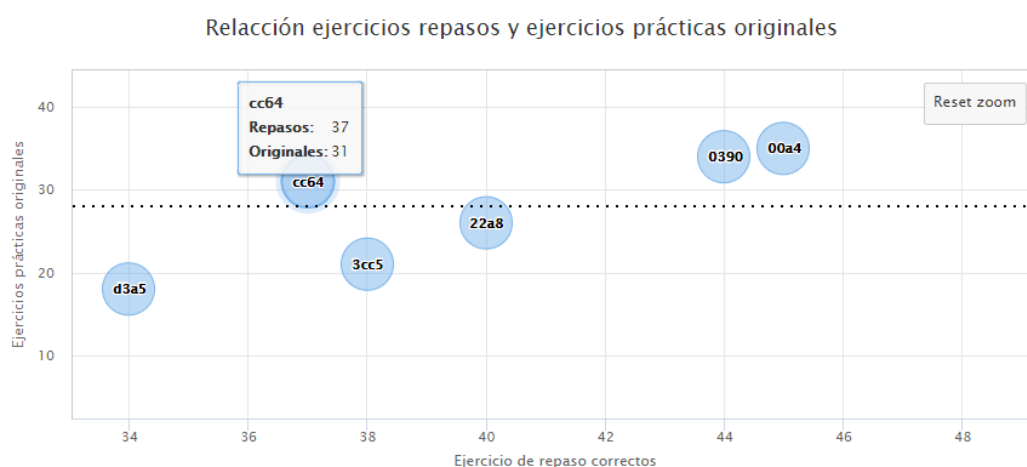


Figura 22. Zoom sobre el gráfico de burbujas

La última opción del menú es “Originales copias” y muestra en un gráfico qué alumnos son copiados y qué otros alumnos les copian (ver figura 23).

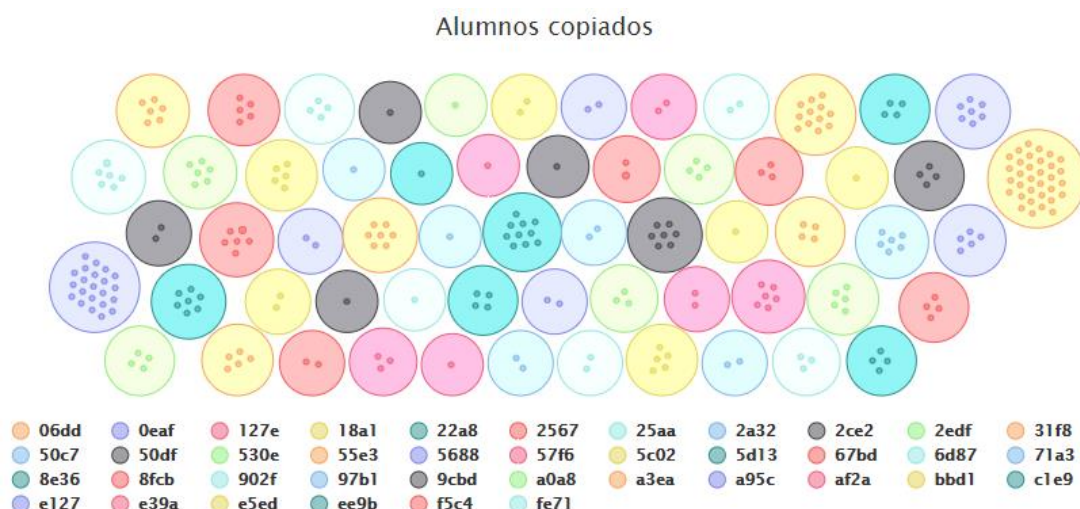


Figura 23. Gráfico de burbujas de los alumnos que son copiados

La gráfica de la figura 23 es una gráfica de burbujas, donde las burbujas grandes corresponden a los alumnos que son copiados, y las burbujas que están dentro de ellas son alumnos que copian a ese alumno. En esta gráfica puede apreciarse qué alumnos copian y tomar alguna medida como por ejemplo situarlos en clase en otras posiciones.



Figura 24. Zoom gráfico burbujas

A través de estas gráficas, los profesores pueden identificar a los alumnos que siguen estrategias perdedoras (como copiar mucho y repasar poco). También puede sugerir a algunos la realización de ejercicios concretos de repaso que les permita mejorar. El objetivo es mejorar el rendimiento en los exámenes intermedios y finales.

Antes de realizar este proyecto los profesores disponían de los datos que les permitían analizar la situación de cada estudiante, pero no disponían del tiempo necesario para analizarlo. La aportación de este proyecto a la enseñanza de la parte práctica de esta asignatura es hacer los análisis accesibles, actualizados y presentados de forma visual.

3.2. Análisis predictivo

En la segunda parte del proyecto se va a realizar un análisis predictivo de las notas finales de los alumnos.

La predicción temprana de las notas que pueden obtener los alumnos resultará de utilidad a los profesores para poder tomar medidas en función de la previsión de éxito de cada uno.

Para desarrollar esta predicción se ha utilizado la herramienta weka, integrada también dentro de la suite Pentaho, para la creación de los modelos predictivos.

3.2.1. Proceso ETL para la predicción del éxito o fracaso

Para la creación del modelo en weka es necesario disponer de un dataset (conjunto de datos), con el cual poder entrenar a dicho modelo.

Para la creación del *dataset* es importante definir cuáles son los atributos que lo van a componer. En este caso, estos atributos son las notas que los alumnos van obteniendo a lo largo del curso en los distintos exámenes que realizan.

Con base en el artículo mencionado anteriormente, no solo las notas de los exámenes son importantes para la predicción de la calificación final, sino que hay una gran correlación entre las copias de los ejercicios de prácticas y los ejercicios de repaso realizados con la nota final del curso, por lo que estos dos datos también forman parte de los atributos del *dataset*.

Para obtener la información mencionada anteriormente ha sido necesario la creación de un proceso ETL.



Figura 25. Trabajo principal ETL

Las tres primeras transformaciones del proceso ETL corresponden con la obtención y detención de copias de los ejercicios de las prácticas, la obtención de los ejercicios de repaso y de las notas de los alumnos. Estos tres pasos son los mismos que los explicados anteriormente en el proceso ETL de la parte de inteligencia de negocio.

La última transformación, “Creación dataset”, es la encargada de obtener los datos de las notas, las copias y los ejercicios de repaso realizados por cada uno de los alumnos, transformar las notas finales en las notas que se van a estimar: aprobado (mayor o igual 6), duda (entre 3.5 y 6) y suspenso (menor o igual 3.5) y almacenar esta información en un fichero *Attribute-Relation File Format* (ARFF).

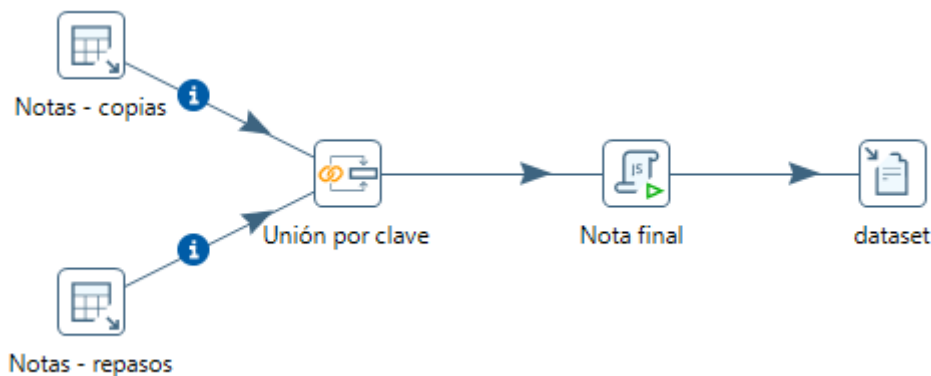


Figura 26. Transformación creación fichero arff

3.2.2. Creación del dataset

Mediante el proceso ETL explicado en el apartado anterior, se han incluido los datos con identificaciones ocultas de los cursos 2014-2015, 2015-2016, 2016-2017 y 2017-2018.

Los datos obtenidos se han almacenado en un fichero cuyo formato es ARFF, este es el fichero nativo que utiliza weka.

Los archivos ARFF deben respetar una estructura determinada, por lo que hay que reestructurar el archivo creado en el proceso ETL.

A continuación, se puede observar la estructura final del *dataset* utilizado para la creación del modelo.


```
@relation notas
```

```
@attribute 'Preaba 1 y 2' numeric  
@attribute 'Prueba 3 a 6' numeric  
@attribute 'Prueba 7 a 9' numeric  
@attribute 'Entrega sql' numeric  
@attribute 'T1' numeric  
@attribute 'T2' numeric  
@attribute 'T3' numeric  
@attribute 'Total test' numeric  
@attribute 'No recuperable' numeric  
@attribute 'Copias' numeric  
@attribute 'Repasos' numeric  
@attribute 'Mayo' {Suspendido,Duda, Aprobado}
```

```
@data
```

```
5.0,6.3,6.5,96.7,2.0,6.0,5.0,4.3333333333,6.2963333333,0,20,Duda  
3.0,0.25,3.5,80.0,2.0,6.0,5.0,4.3333333333,2.9133333333,11,14,Suspendido  
4.0,3.25,5.0,100.0,4.0,4.0,5.0,4.3333333333,4.7133333333,7,18,Duda  
3.0,0.0,6.0,93.3,8.0,4.0,8.35,6.7833333333,4.0113333333,5,17,Aprobado  
5.0,1.75,9.0,96.7,4.0,2.0,5.0,3.6666666667,5.5736666667,15,0,Duda  
3.0,0.5,5.0,96.7,2.0,2.0,8.35,4.1166666667,3.6186666667,11,10,Suspendido  
5.0,4.3,1.5,76.7,8.0,8.0,6.68,7.56,4.179,4,14,Suspendido  
3.0,2.6,1.5,70.0,4.0,0.0,3.34,2.4466666667,2.7366666667,3,6,Suspendido  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0,0,Suspendido  
5.0,3.7,1.5,96.7,6.0,10.0,6.68,7.56,4.187,6,33,Suspendido  
6.0,5.3,3.0,100.0,0.0,4.0,3.34,2.4466666667,4.8606666667,6,73,Duda  
9.0,9.25,10.0,100.0,4.0,8.0,8.35,6.7833333333,9.2783333333,2,128,Aprobado  
5.0,6.25,6.0,96.7,6.0,8.0,3.34,5.78,6.265,1,0,Aprobado  
5.0,1.8,0.5,93.3,6.0,6.0,3.34,5.1133333333,2.9803333333,11,0,Suspendido  
4.0,7.8,2.0,80.0,6.0,4.0,3.34,4.4466666667,5.0206666667,8,49,Suspendido  
6.0,4.75,10.0,100.0,2.0,0.0,3.34,1.78,6.858,10,23,Duda  
5.0,2.0,6.0,100.0,6.0,4.0,6.68,5.56,4.916,18,27,Suspendido  
10.0,7.7,10.0,90.0,8.0,8.0,6.68,7.56,8.92,5,85,Aprobado  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0,0,Suspendido  
5.0,0.25,4.5,83.3,4.0,2.0,6.68,4.2266666667,3.5756666667,7,18,Duda
```

Figura 27. Dataset archivo arff

En la imagen anterior se aprecia que la estructura de los archivos ARFF se divide en las siguientes tres partes [7]:

1. Cabecera, en la que se define el nombre de la relación a través de @relation.
2. Declaración de atributos. Estos atributos se declaran a través de la siguiente estructura: @attribute 'Nombre del atributo' tipo. Los tipos de los atributos pueden ser numéricos, enteros, fechas, cadenas de texto o conjuntos de datos.
3. Por último, se encuentra la sección de los datos, introducida por la anotación @data.

3.2.3. Creación del modelo

Como se ha explicado ya, el objetivo es predecir la nota desde una etapa temprana de la asignatura, por lo que se van a crear dos modelos diferentes para poder predecir la nota en dos fases del curso.

Se había observado en el estudio estadístico realizado por los profesores [1] que la nota que obtienen los alumnos en el segundo examen de evaluación continua, que corresponde a las sesiones de laboratorio 3, 4 y 5, tiene una fuerte correlación positiva con la nota final. Por tanto, en el primer modelo que se ha creado se utilizaron como atributos las notas de los exámenes realizados hasta ese momento, además de las copias y los ejercicios de repaso.

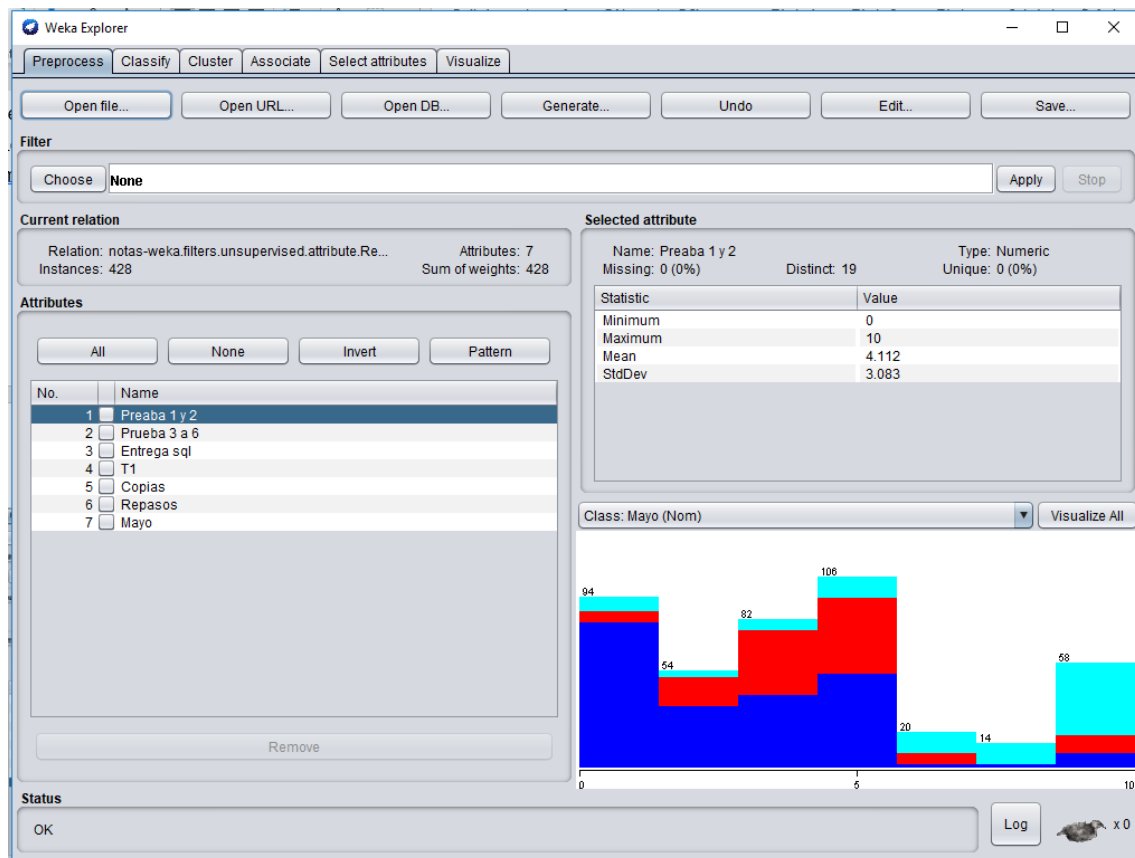


Figura 28. Weka creación del primer modelo

A continuación, se pusieron a prueba diferentes algoritmos de clasificación para obtener los mejores resultados posibles. El modo de entrenamiento utilizado fue el de validación cruzada (Cross-validation). Los resultados que se obtuvieron son los que figuran a continuación.

RandomForest

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      412           96.2617 %
Incorrectly Classified Instances    16           3.7383 %
Kappa statistic                     0.9392
Mean absolute error                  0.1154
Root mean squared error              0.19
Relative absolute error              27.9461 %
Root relative squared error          41.8385 %
Total Number of Instances          428
```

J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      344           80.3738 %
Incorrectly Classified Instances     84           19.6262 %
Kappa statistic                     0.6809
Mean absolute error                  0.1327
Root mean squared error              0.3205
Relative absolute error              32.1482 %
Root relative squared error          70.5657 %
Total Number of Instances          428
```

REPTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      324           75.7009 %
Incorrectly Classified Instances    104           24.2991 %
Kappa statistic                     0.6005
Mean absolute error                  0.2131
Root mean squared error              0.3529
Relative absolute error              51.6201 %
Root relative squared error          77.692 %
Total Number of Instances          428
```

RandomTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      402           93.9252 %
Incorrectly Classified Instances     26           6.0748 %
Kappa statistic                     0.9016
Mean absolute error                  0.0434
Root mean squared error              0.2012
Relative absolute error              10.524 %
Root relative squared error          44.2981 %
Total Number of Instances          428
```

Se realizaron pruebas con otros algoritmos de clasificación y con distintas configuraciones de estos, obteniéndose peores resultados que con los cuatro algoritmos que se acaban de presentar. Por tanto, el modelo que dio mejores resultados es el del algoritmo RandomForest.

Para crear el segundo modelo los atributos utilizados son las notas de los exámenes de la evaluación continua, la nota final de las entregas de las practicas, los test y el número de ejercicios de repaso y de las copias realizadas.

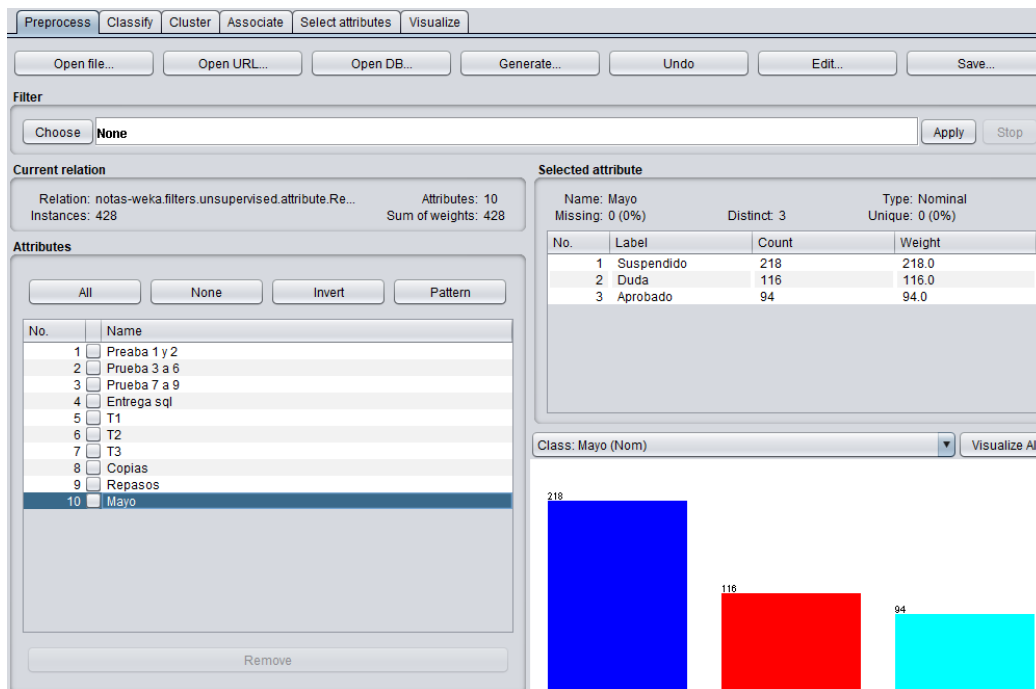


Figura 29. Weka creación segundo modelo

En este modelo también se usó como método de entrenamiento la validación cruzada y se probaron distintos algoritmos de clasificación para encontrar el más adecuado.

J48

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      363           84.8131 %
Incorrectly Classified Instances    65           15.1869 %
Kappa statistic                     0.7527
Mean absolute error                 0.0995
Root mean squared error             0.2869
Relative absolute error             24.1064 %
Root relative squared error         63.1665 %
Total Number of Instances          428

```

Random forest

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      414           96.729 %
Incorrectly Classified Instances    14           3.271 %
Kappa statistic                     0.947
Mean absolute error                 0.0996
Root mean squared error            0.1747
Relative absolute error             24.1337 %
Root relative squared error        38.4576 %
Total Number of Instances         428
```

REPTree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      340           79.4393 %
Incorrectly Classified Instances    88           20.5607 %
Kappa statistic                     0.665
Mean absolute error                 0.1703
Root mean squared error            0.3299
Relative absolute error            41.2575 %
Root relative squared error        72.6187 %
Total Number of Instances         428
```

DecisionStump

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      294           68.6916 %
Incorrectly Classified Instances   134           31.3084 %
Kappa statistic                     0.4761
Mean absolute error                 0.2844
Root mean squared error            0.3776
Relative absolute error            68.8914 %
Root relative squared error        83.1381 %
Total Number of Instances         428
```

Nuevamente se obtienen los mejores resultados con el algoritmo *Random forest*. Una vez creados los dos modelos, se exportaron para utilizarlos en la aplicación web.

3.2.4. Implementación del análisis predictivo

La implementación del modelo se programa en la parte del back-end de la aplicación y para ello, el primer paso para poder utilizar las opciones

proporcionadas por weka es incluir la siguiente dependencia en el fichero pom.xml[8]

```
<dependency>
  <groupId>nz.ac.waikato.cms.weka</groupId>
  <artifactId>weka-stable</artifactId>
  <version>3.8.0</version>
</dependency>
```

Seguidamente, se muestra el proceso para poder predecir una nota. Lo primero que hay que hacer es cargar el modelo que se ha creado con weka.

```
InputStream i = getClass().getClassLoader().getResourceAsStream( name: "ModeloRandomForest.model");
Classifier classifier = (Classifier) weka.core.SerializationHelper.read(i);
```

Una vez cargado el modelo, hay que preparar el *dataset* en el que se encuentra la instancia que se quiere clasificar. El siguiente paso es la declaración de los atributos que se van a incluir en la instancia a predecir. En este caso, los atributos son los mismos que se han utilizado en la creación del modelo.

```
ArrayList<String> clasificacion = new ArrayList<String>();
clasificacion.add("Suspendido");
clasificacion.add("Duda");
clasificacion.add("Aprobado");

ArrayList<Attribute> atts = new ArrayList<>( initialCapacity: 10);
atts.add(new Attribute( attributeName: "Preaba 1 y 2"));
atts.add(new Attribute( attributeName: "Prueba 3 a 6"));
atts.add(new Attribute( attributeName: "Prueba 7 a 9"));
atts.add(new Attribute( attributeName: "Entrega sql"));
atts.add(new Attribute( attributeName: "T1"));
atts.add(new Attribute( attributeName: "T2"));
atts.add(new Attribute( attributeName: "'T3'"));
atts.add(new Attribute( attributeName: "Copias"));
atts.add(new Attribute( attributeName: "Repasos"));
atts.add(new Attribute( attributeName: "Mayo", clasificacion));
```

A la hora de inicializar la instancia se incluyen en el constructor los atributos declarados anteriormente y los valores necesarios para la clasificación, en este caso, las notas, copias y repasos de un determinado alumno.

```

Instances ins_data = new Instances( name: "Notas", atts, capacity: 0);
Instance inst = new DenseInstance(ins_data.numAttributes());
inst.setValue( i: 0, nota.pruebaSesion1 == null ? 0 : nota.pruebaSesion1);
inst.setValue( i: 1, nota.pruebaSesion2 == null ? 0 : nota.pruebaSesion2);
inst.setValue( i: 2, nota.pruebaSesion3 == null ? 0 : nota.pruebaSesion3);
inst.setValue( i: 3, nota.entregasSql == null ? 0 : nota.entregasSql);
inst.setValue( i: 4, nota.test1 == null ? 0 : nota.test1);
inst.setValue( i: 5, nota.test2 == null ? 0 : nota.test2);
inst.setValue( i: 6, nota.test3 == null ? 0 : nota.test3);
inst.setValue( i: 7, nota.numCopias == null ? 0 : nota.numCopias);
inst.setValue( i: 8, nota.numRepasos == null ? 0 : nota.numRepasos);
inst.setValue( i: 9, Utils.missingValue());

```

Como se puede apreciar, el último valor (Utils.missingValue()) corresponde con la calificación de mayo a predecir. Por último, se utiliza el modelo cargado anteriormente para obtener el resultado deseado.

```

double result = classifier.classifyInstance(ins_data.firstInstance());
String notaMyoEstimada = clasificacion.get((int) result);

```

Como se ha explicado con anterioridad, se dispone de dos modelos, por lo que en la aplicación, dependiendo del momento del curso en el que se desee realizar la predicción de las notas, se debe elegir el modelo apropiado.

3.2.5. Análisis predictivo en la aplicación

La parte de análisis predictivo se corresponde con el último menú de la aplicación. Esta pantalla (ver figura 30) está compuesta por un filtro en la parte superior y una tabla en la parte central, en la que se muestran los datos.



Figura 30. Diseño menú predicción

Los filtros permiten obtener la nota final estimada de un determinado alumno (ver figura 32).

Alumno		Nota final		Filtrar
18a1	▼	Ninguno	▼	

Nombre	Sesiones 1 y 2	Sesiones 3 a 6	Sesiones 7 a 9	sql	Test 1	Test 2	Test 3	Total test	No recuperable	Nota final estimada
18a1	5.00	6.75	7.00	92.50	0.00	4.00	5.01	3.00	5.73	Duda

Figura 32. Predicción nota final por alumno

O todos los alumnos que van a obtener una determinada nota final (aprobado, duda, suspendido) como se ve en la figura 33.

Alumno		Nota final		Filtrar
Ninguno	▼	Aprobado	▼	

Nombre	Sesiones 1 y 2	Sesiones 3 a 6	Sesiones 7 a 9	sql	Test 1	Test 2	Test 3	Total test	No recuperable	Nota final estimada
00a4	9.00	8.75	4.50	100.00	4.00	10.00	10.00	8.01	7.38	Aprobado
0eaf	5.00	9.00	9.00	100.00	8.00	6.00	8.35	7.45	7.96	Aprobado
127e	10.00	9.50	8.00	100.00	6.00	6.00	6.68	6.23	8.37	Aprobado
22a8	4.00	6.25	8.50	90.00	2.00	4.00	10.00	5.34	5.89	Aprobado
2a32	8.00	6.25	10.00	100.00	4.00	4.00	5.01	4.34	7.91	Aprobado
2ce2	10.00	10.00	10.00	100.00	10.00	10.00	8.35	9.45	9.95	Aprobado
39e6	10.00	6.50	10.00	100.00	10.00	10.00	8.35	9.45	8.83	Aprobado

« Previous **1** 2 3 4 Next »

Figura 33. Predicción nota final por grupo

4. CONCLUSIONES

Este proyecto se ha finalizado logrando alcanzar los dos objetivos que se habían propuesto inicialmente. El primero son la realización de cuadros de mando mediante aplicación de técnicas de inteligencia de negocio. Los cuadros de mando permiten a los profesores realizar un seguimiento de los repasos y las copias que realizan los estudiantes observando la situación del grupo y personal de los alumnos. El segundo objetivo era la realización de un módulo para el análisis predictivo del éxito de los estudiantes en función de las notas que van obteniendo en la evaluación continua y de su actividad en laboratorios y repasos.

Este proyecto tiene una proyección en el trabajo real de los profesores, ya que se trata de una herramienta que puede resultarles útil para tener una idea actualizada y de fácil acceso sobre la evolución de los estudiantes durante el curso. Esta información les permitirá tomar conciencia de lo que ocurre y poder aplicar nuevas estrategias docentes adaptadas a cada tipo de estudiante. El fin último de todo el trabajo es contribuir a la mejora de la enseñanza/aprendizaje y que esta se refleje en la mejora de los resultados académicos de los estudiantes.

En este trabajo se han podido poner en práctica diferentes técnicas y tecnologías adquiridas estudiando las asignaturas del máster en el que se encuadra este proyecto. Por tanto, se ha logrado el objetivo personal de profundizar y afianzar estos conocimientos.

Bajo mi punto de vista la aplicación de estas técnicas es exportable a otras asignaturas de la titulación y este proyecto puede servir como referencia a futuros desarrolladores que necesiten encontrar soluciones parecidas. Haber utilizado software gratuito, como la suite Pentaho, le aporta mayor accesibilidad.

Obviamente este trabajo puede ser extendido con nuevas gráficas o adaptar el predictor. A partir del curso que viene se podrá poner a prueba y apreciar en qué medida es de utilidad o conviene ajustar los elementos visuales o la capacidad del predictor.

5. BIBLIOGRAFÍA

- [1] Domínguez, C., Jaime, A., Heras, J., & García-Izquierdo, F. J. (2019). The Effects of Adding Non-Compulsory Exercises to an Online Learning Tool on Student Performance and Code Copying. *ACM Transactions on Computing Education (TOCE)*, 19(3), 16.
- [2] <https://community.hitachivantara.com/docs/DOC-1009855-data-integration-kettle>
- [3] <https://angular.io/>
- [4] <https://www.highcharts.com>
- [5] <https://www.npmjs.com/package/highcharts-angular>
- [6] <https://api.highcharts.com/highcharts/>
- [7] <https://txikiboo.wordpress.com/2014/01/16/archivos-arff-weka/>
- [8] <http://www.emaraic.com/blog/weka-java-example>